

AMENDMENT TO THE CLAIMS

Please accept the following amended claims. Claims 1, 3, 4, 10, 11, 13, 14 and 20 have been amended. Claims 2 and 12 have been cancelled. (All claims listed)

1 1. (Currently amended) A method of establishing thread priority in a single processor
2 comprising:
3 assigning a value in memory to indicate which of a plurality of threads executed by said
4 single processor has a higher priority; and
5 allocating a resource between said plurality of threads depending on a priority assigned to
6 each thread.

1 2. (Cancelled)

1 3. (Currently amended) The method of claim 21 wherein in said allocating step, a first
2 thread is given greater access to a the resource than other threads when said first thread is
3 assigned a higher priority than said other threads.

1 4. (Currently amended) The method of claim 21, wherein in said allocating step, the other
2 threads are given greater access to the resource than the first thread when said first thread is
3 assigned a higher priority than the other threads and is not using said resource.

1 5. (Original) The method of claim 3 wherein said resource is a unit in a processor system.

1 6. (Original) The method of claim 5 wherein said resource is a decode unit.

1 7. (Original) The method of claim 6 further comprising:

2 providing instructions from a first thread to a first queue;

3 providing instructions from a second thread to a second queue;

4 supplying a first number of instructions to said decode unit from said first queue;

5 supplying a second number of instructions to said decode unit from said second queue;

6 selecting said first and second numbers based on said value in memory.

1 8. (Original) The method of claim 3 wherein said resource is a bus.

1 9. (Original) The method of claim 8 further comprising:

2 providing bus requests from a first thread to a first queue;

3 providing bus requests from a second thread to a second queue;

4 servicing a first number of bus requests from the first queue;

5 servicing a second number of bus requests from said second queue; and

6 selecting said first and second numbers based on said value in memory.

1 10. (Currently amended) A method of establishing thread priority in a single processor
2 comprising:

3 assigning a value in an APIC TPR register for a thread via execution of operating system
4 code to indicate which of a plurality of threads executed by said single processor has a higher
5 priority; and

6 allocating a resource between said plurality of threads depending on a priority assigned to
7 each thread.

1 11. (Currently amended) An apparatus for establishing thread priority in a single processor
2 comprising:

3 a memory to store a value to indicate which of a plurality of threads to be executed by
4 said single processor has a higher priority-; and

5 a resource allocated between said plurality of threads depending on a priority assigned to
6 each thread in said memory.

1 12. (Cancelled)

1 13. (Currently amended) The apparatus of claim ~~42~~11 wherein a first thread is given greater
2 access to a the resource than other threads when said first thread is assigned a higher priority
3 than said other threads.

1 14. (Currently amended) The apparatus of claim ~~42~~11 wherein the other threads are given
2 greater access to the resource than the first thread when said first thread is assigned a higher
3 priority than the other threads and is not using said resource.

1 15. (Original) The apparatus of claim 13 wherein said resource is a unit in a processor
2 system.

1 16. (Original) The apparatus of claim 15 wherein said resource is a decode unit.

1 17. (Original) The apparatus of claim 16 further comprising:

2 a first queue to store instructions from a first thread;

3 a second queue to store instructions from a second thread;

4 control logic coupled to said first and second queues and said decode unit, said control

5 logic to permit a first number of instructions to be transferred to said decode unit then a second

6 number of instructions to be transferred to said decode unit, said first and second numbers being

7 selected based on said value in memory.

1 18. (Original) The apparatus of claim 13 wherein said resource is a bus.

1 19. (Original) The apparatus of claim 18 further comprising:

2 a bus unit including

3 a first queue storing bus requests from a first thread;

4 a second queue storing bus requests from a second thread;

5 control logic coupled to said first and second queues, said control logic to control

6 servicing of a first number of bus requests from the first queue and a second number of bus

7 requests from said second queue, said first and second number being selected based on said value

8 in memory.

1 20. (Currently amended) An apparatus for establishing thread priority in a single processor

2 comprising:

3 an APIC TPR register for a thread wherein execution of operating system code causes a
4 value to be stored in said register to indicate which of a plurality of threads to be executed by
5 said single processor has a higher priority; and
6 a resource allocated between said plurality of threads depending on a priority assigned to
7 each thread in said memory.